



---

# 中国电信物联网开放平台\_NB-IoT 业务 对接指导书

---

(V1)

二〇一八年一月

概述

本文档针对 NB-IoT 业务，从准备工作、自定义设备模型、编解码插件开发、应用接入平台、设备接入平台及消息上下行等方法流程进行了介绍。

本文档提供了 NB-IoT 业务设备接入平台的指导。


读者对象

- 本文档主要适用于以下工程师：
- 开发工程师
  - 技术支持工程师
  - 维护工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	用于警示紧急的危险情形，若不可避免，将会导致人员死亡或严重的人身伤害。
 警告	用于警示潜在的危险情形，若不可避免，可能会导致人员死亡或严重的人身伤害。
 小心	用于警示潜在的危险情形，若不可避免，可能会导致中度或轻微的人身伤害。
 注意	用于传递设备或环境安全警示信息，若不可避免，可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “注意”不涉及人身伤害。

符号	说明
 说明	用于突出重要/关键信息、最佳实践和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

本规范由中国电信集团公司提出并归口。

本规范起草单位：中国电信物联网开放实验室

中国电信

# 目 录

前 言.....	ii
目 录.....	1
1 概述.....	1
1.1 功能组网介绍 .....	1
1.2 整体业务介绍 .....	1
1.3 业务测试接入流程 .....	2
2 准备工作.....	4
2.1 获取测试环境账号 .....	4
2.2 创建 APP 应用 .....	7
2.3 NB-IoT 参数设置.....	8
3 开发 Profile .....	10
3.1 Profile 介绍 .....	10
3.2 登录开发者 Portal.....	10
3.3 开发 Profile .....	10
4 开发编解码插件并安装 .....	13
4.1 编解码插件介绍 .....	13
4.2 开发者 Portal 在线插件开发.....	13
4.2.1 登录开发者 Portal.....	13
4.2.2 在线完成插件开发 .....	13
4.3 离线开发编解码插件 .....	16
4.3.1 开发编解码插件 .....	16
4.3.2 编解码插件质检 .....	16
4.3.3 对插件包进行离线签名 .....	16
4.3.4 上传公钥及签名后的插件包.....	18
5 应用接入平台.....	19
5.1 登陆应用 .....	19
5.2 订阅 .....	19
5.3 注册设备 .....	19

5.3.1 有应用服务器设备注册 .....	19
5.3.2 无应用服务器设备注册 .....	20
5.4 设置设备信息 .....	20
<b>6 设备接入平台.....</b>	<b>22</b>
6.1 设备接入 .....	22
6.2 设备上线 .....	23
6.2.1 真实设备 .....	23
6.2.2 模拟设备 .....	23
<b>7 业务数据上报和业务消息下发流程 .....</b>	<b>25</b>
7.1 接收设备上报的数据 .....	25
7.2 向设备发送消息 .....	27
<b>8 命令下发.....</b>	<b>1</b>
8.1 命令立即下发 .....	1
8.2 命令缓存下发 .....	3
<b>9 更多功能.....</b>	<b>4</b>
<b>10 测试环境测试.....</b>	<b>4</b>
10.1 测试申请 .....	4
10.2 开放实验室测试 .....	4
<b>11 生产环境接入.....</b>	<b>5</b>
11.1 账号申请 .....	5
11.2 生产环境接入 .....	5
<b>12 参考文档.....</b>	<b>6</b>
12.1 中国电信物联网开放平台编解码库开发与升级指南 .....	6
12.2 中国电信物联网开放平台 API 参考 1.3.1 .....	6
12.3 中国电信物联网开放平台北向 API Java SDK 说明文档 .....	6

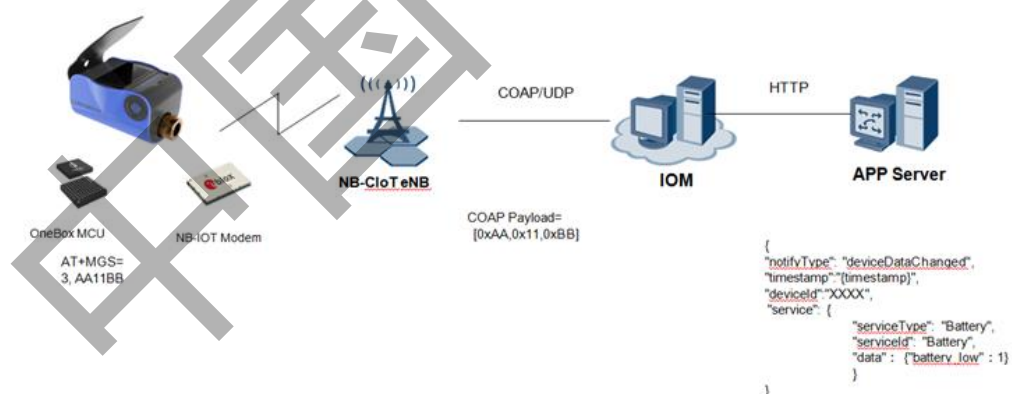
# 1 概述

## 1.1 功能组网介绍

在设备侧，智能设备采用了 NB-IOT 芯片，它和中国电信物联网开放平台之间通过 CoAP 协议通讯。CoAP 报文的 payload 里是设备的应用数据。

应用 server 通过 http/https 协议和平台通讯，通过调用平台的开放 API 来控制设备，平台把设备上报的数据推送给应用服务器。平台支持对设备数据进行协议解析，转换成标准的 json 格式数据。

图1-1 组网图



## 1.2 整体业务介绍

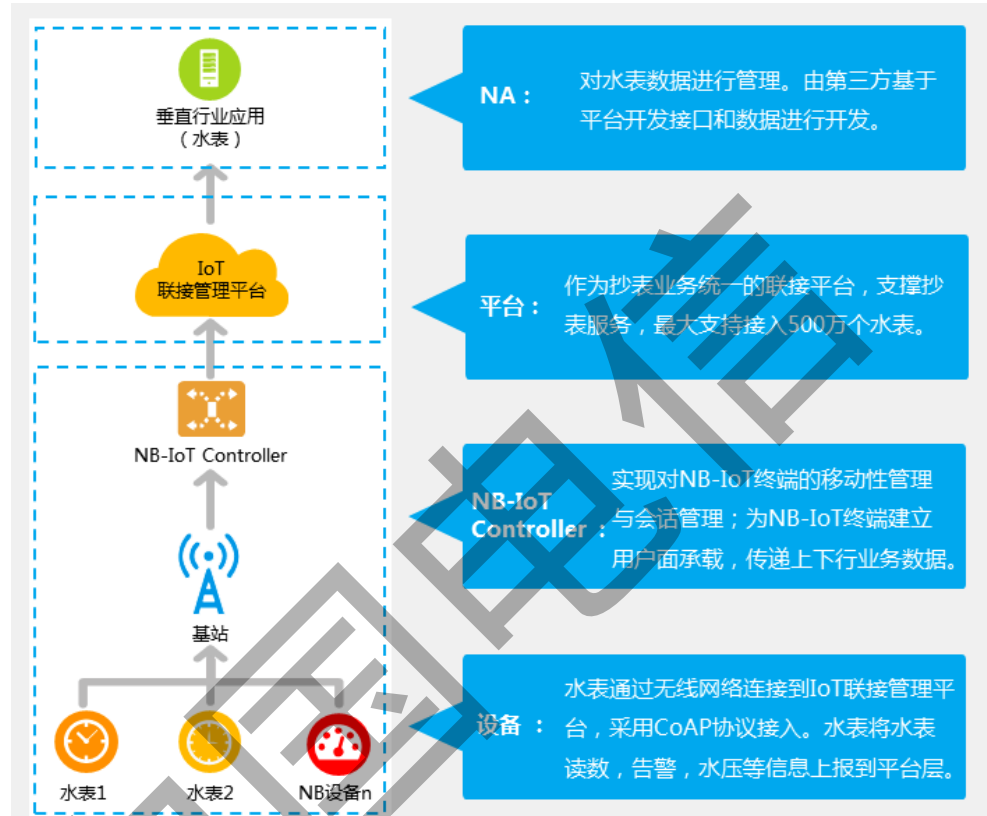
NB-IOT 主要包含 NB 设备、NB-IOT 网络、中国电信物联网开放平台、垂直行业应用：

**NB-IOT 设备：**通过无线网络连接到中国电信物联网开放平台，采用 CoAP 协议接入，如水表、燃气表等。

**NB-IOT 网络：**实现对 NB-IoT 终端的移动性管理与会话管理；为 NB-IoT 终端建立用户面承载，传递上下行业务数据。

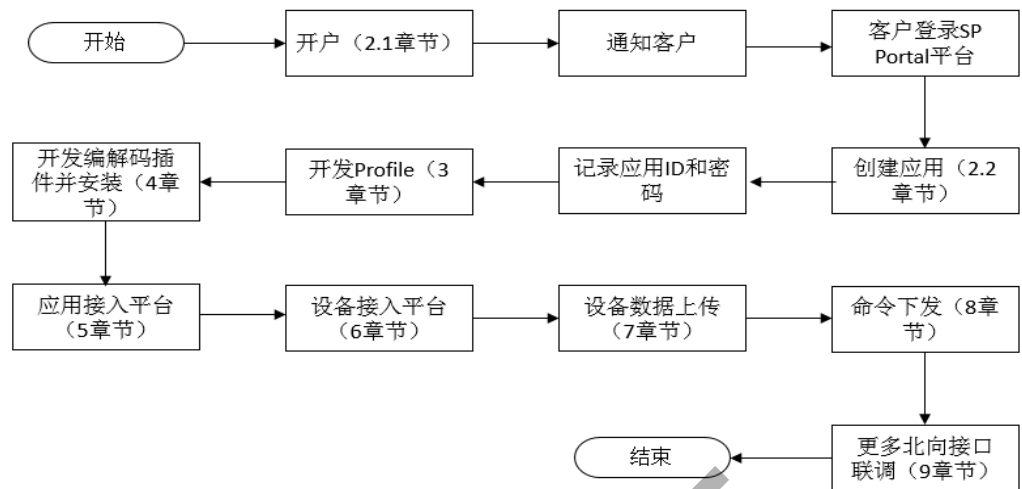
**中国电信物联网开放平台：**实现对各种 NB-IoT 设备数据的统一管理，同时向第三方应用系统开放接口，让各种应用能快速构建自己的物联网业务。

**垂直行业应用：**实现对 NB 设备的业务管理，包括业务发放、业务控制和呈现等，由第三方基于中国电信物联网开放平台开放接口进行开发。



## 1.3 业务测试接入流程

为客户接入测试便捷化，制定了如下的测试接入流程：





# 2 准备工作

## 2.1 获取测试环境账号

### 2.1.1 微信公众号申请

客户或电信政企经理登录“天翼物联产业联盟”微信公众号填写开放平台测试账号申请。

图2-1 查找微信公众号并进入

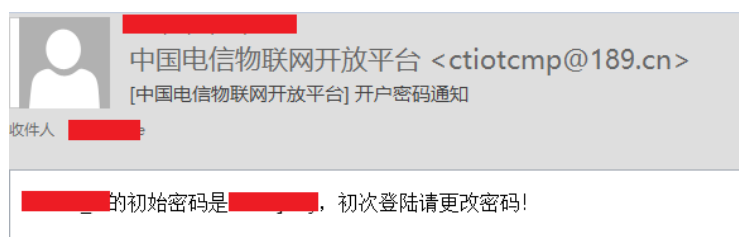


图2-2 点击联盟服务选择“平台联调服务申请”



图2-3 完成表格填写并提交

图2-4 账号申请通过后，收到账号和初始密码



----结束

## 2.1.2 网站申请

客户或者电信政企经理登录“天翼物联产业联盟”网站（www.tianyijiot.com）填写开放平台测试账号申请。

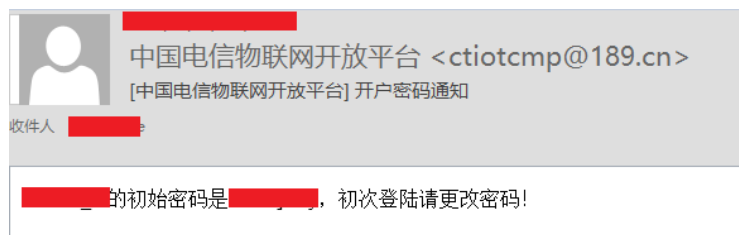
图2-5 登录网站选择“平台联调服务申请”



图2-6 完成表格填写并提交

服务申请	
单位名称*	<input type="text" value="请填写您的单位名称"/>
联络人*	<input type="text" value="请填写与联络人"/>
邮箱*	<input type="text" value="请填写写邮箱"/>
电话*	<input type="text" value="请填写写电话"/>
物联网应用简述*	<div>简述物联网产品功能如：路灯远程与数据上报，蔬菜大棚监控...</div> <div></div>

图2-7 账号申请通过后，收到账号和初始密码

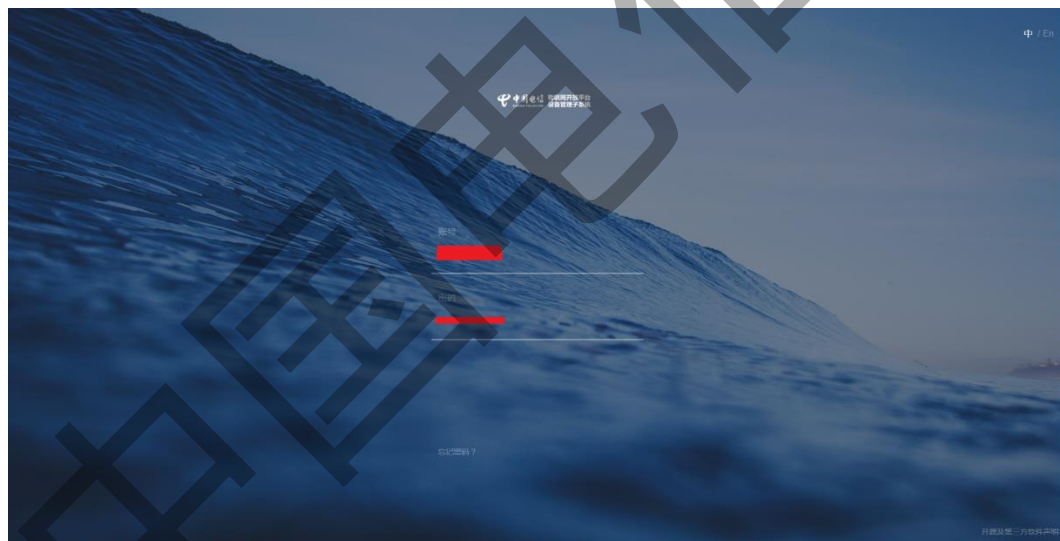


----结束

## 2.2 创建 APP 应用

登录中国电信物联网开放平台对接测试环境 SP Portal (<https://180.101.147.135:8843>)。初次登录需要修改初始密码。

图2-8 SP Portal



选择【应用管理】->【应用】点击右上角“+创建应用”。

典型的 NB-IoT 应用的参数设置可参考如下设置：

所属行业：公用事业(NB-IoT)

关联 API 包：基础 API 包和公用事业(NB-IoT) API 包

平台能力：必须设置为规则引擎

数据存储时间：7（此为默认值，可修改）。超过了数据存储时间的设备历史数据将不再保存，即 1 条设备历史数据只能在平台上保存 7 天（以默认值为例）。

短信服务器：与规则引擎配合使用，可以在指定条件下通过短信服务器发送短信给指定人员。服务器可以选择平台已经支持的服务器之一，也可以选择第三方服务器。

邮件服务器：与规则引擎配合使用，可以在指定条件下通过邮件服务器发送邮件给指定人员。服务器可以选择平台已经支持的服务器之一，也可以选择第三方服务器。

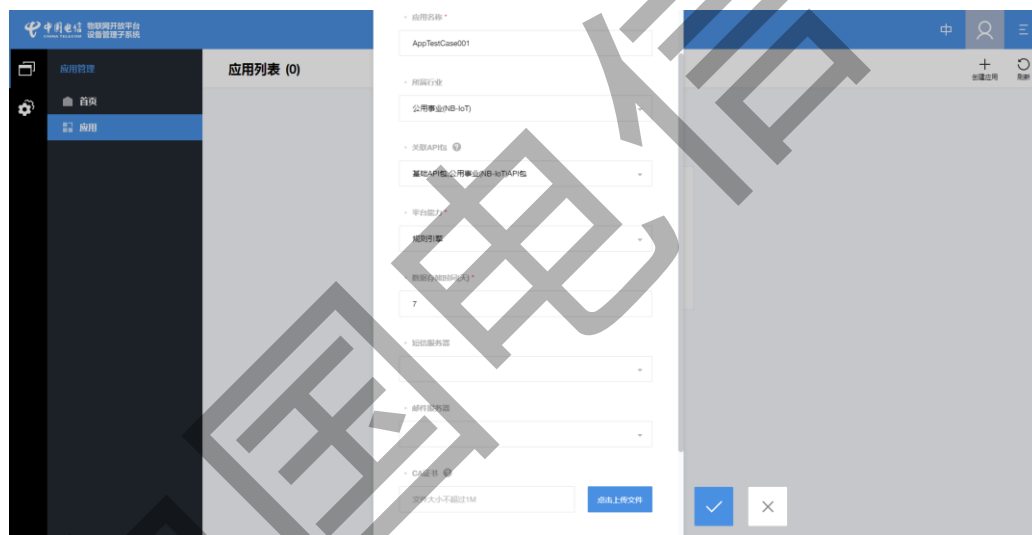
**CA 证书：**用于消息推送场景，需要上传应用服务器的 CA 证书。如果订阅的回调地址是 HTTPS 地址，就需要上传证书。证书文件限制为不超过 1M 的 PEM 格式文件；中国电信物联网开放平台预制了 7 个 CA 机构颁发的证书，如果对接的第三方应用的证书在这 7 个之中，则不需要上传到平台，否则需要在 IoT 中进行上传。7 个证书名称分别为 CA baltimore、digicert、entrust、geotrust、globalsign、thawte、verisign。

#### 说明

消息推送场景下，平台是客户端，应用是服务端，平台调用应用的接口给应用推送消息。应用需要实现消息推送的接口（只实现应用需要用到的消息推送接口即可）。

当第三方应用通过 HTTP 对接中国电信物联网开放平台时，不需要上传 CA 证书，但是需要把接收消息 IP 地址（即回调 IP 地址）发给平台支撑人员，将 IP 添加到白名单中。

图2-9 创建应用



创建 APP 成功后，得到应用 ID 和应用密钥，请保存。

-----结束

## 2.3 NB-IoT 参数设置

点击已经创建好的应用，并对 NB-IoT 参数进行设置。

图2-10 已创建应用

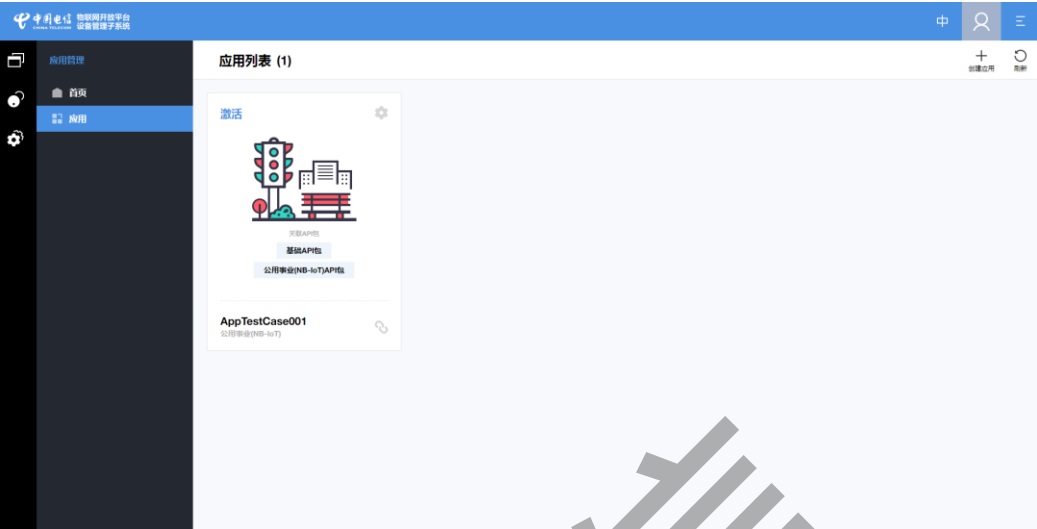
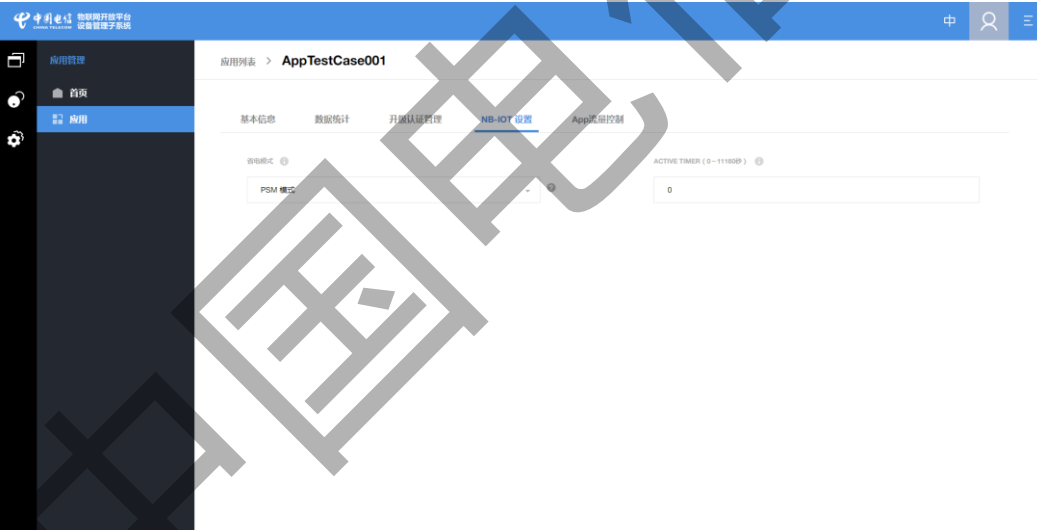


图2-11 NB-IoT 设置

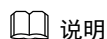


NB-IoT 设置下的省电模式，应与受理的 APN 对应的模式保持一致。平台根据设置的省电模式及参数来决定离线命令下发给终端的方式。应用下发控制命令时可以选择立即下发或缓存下发，但在不同省电模式下，其表现形式不同。

省电模式选择为 PSM 模式：当命令为缓存下发时，平台将收到的下发命令缓存，等设备上报数据时，按顺序将缓存的命令全部下发；当命令为立即下发时，平台将收到的下发命令立即下发，如果此时，设备处于 PSM 态，设备将不能收到下发命令，设备处于激活态或空闲态将能否收到下发命令。

省电模式选择 DRX 模式：当命令为缓存下发时，平台将收到的下发命令缓存，按顺序将缓存的命令全部下发；当命令为立即下发时，平台将收到的下发命令立即下发，设备能够收到命令。

省电模式选择 eDRX 模式：当命令为缓存下发时，平台将收到的下发命令缓存，按顺序将缓存的命令全部下发；当命令为立即下发时，平台将收到的下发命令立即下发，设备能够收到命令。



说明

当省电模式选择 eDRX 模式时，eDRX 周期的设置需要与网络侧的一致，请联系中国电信物联网开放平台支持人员获取参数配置数据。

----结束

## 3 开发 Profile

---

### 3.1 Profile 介绍

设备的 Profile 文件是用来描述一款设备是什么、能做什么以及如何控制该设备的文件。每款设备都需要一个 Profile 文件。

一个 Profile 文件的内容主要包括两部分，一部分是设备的基础属性信息，如 `manufacturerId`（厂商 ID）、`manufacturerName`（厂商名字）、`deviceType`（设备类型）、`model`（设备型号）、`protocolType`（协议类型）等；另一部分是设备的服务信息，这一部分定义了设备的业务数据，包括设备上报的上行数据和厂商服务器下发给设备的下行数据。

### 3.2 登录开发者 Portal

开发者 Portal 支持在线开发 Profile；对开发完成的 Profile 也支持导出和导入。

请使用 [章节 2.1](#) 获取到的账号和密码登录开发者 Portal。如果初始密码已经修改，请用修改后的密码登录。

开发者 Portal 地址：<https://180.101.147.208:8093>

### 3.3 开发 Profile

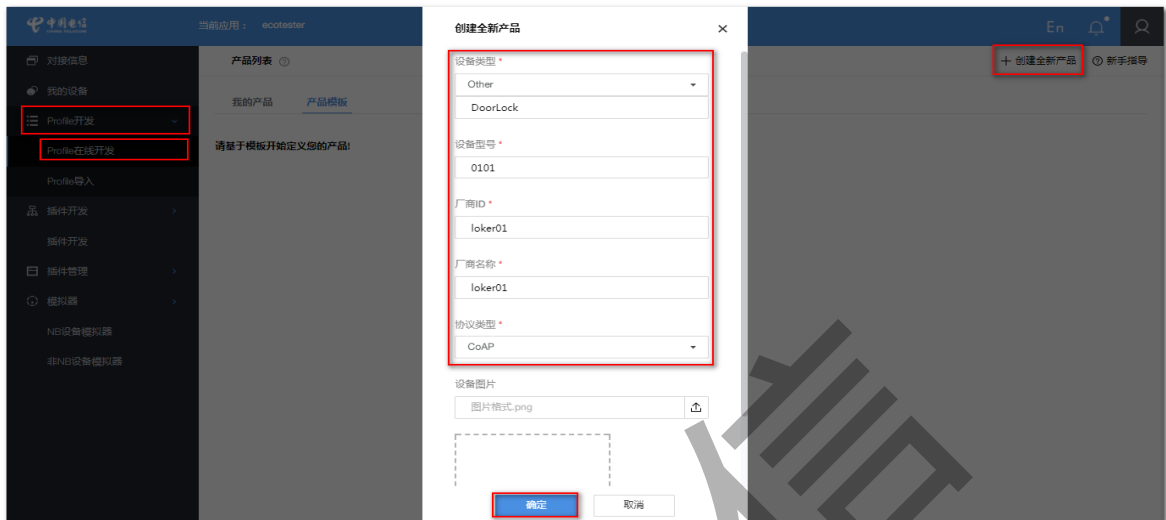
请参考开发者 Portal 使用指导视频完成 Profile 开发。

开发者 Portal 使用指导视频下载地址：

<http://www.tianyiit.com/downloadZone/downloadZones.html>

大致开发步骤如下：


登录开发者 portal->Profile 开发->Profile 在线开发->自定义产品->创建全新产品->填写设备类型、设备型号、厂商 ID、厂商名称、协议类型，点击确定（如下图所示）



再根据设备的功能和业务场景添加服务，每个服务下可以增加属性（上报的数据）和命令（下发的命令，命令必须有字段），如下图所示：



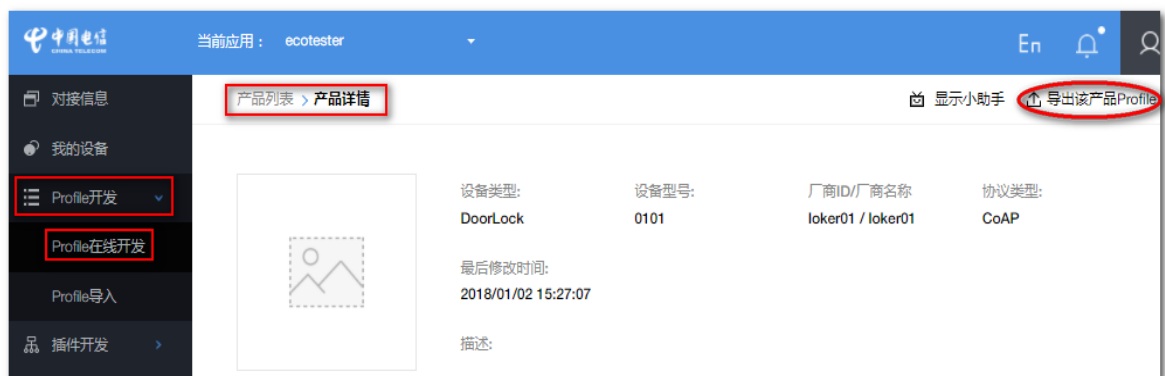
一个设备可以有多个服务：



说明



在线开发的 profile 可以导出成 profile 资源包，然后上传到其他 IoT 平台上使用。导出方法如下图所示：



# 4 开发编解码插件并安装

## 4.1 编解码插件介绍

NB-IoT 设备和中国电信物联网开放平台之间采用 CoAP 协议通讯，CoAP 消息的数据为应用层数据，应用层数据的格式由设备厂商自行定义。由于 NB-IoT 设备对省电要求通常较高，所以应用层数据一般采用二进制格式。应用层数据在中国电信物联网开放平台做协议解析时，会转换成统一的 json 格式，以方便应用服务器使用。要实现二进制消息与 json 格式消息的转换功能，中国电信物联网开放平台需要使用设备厂商提供的编解码插件。一款设备对应一个编解码插件。

开发编解码插件有两种方式：

方式一：开发者 Portal 在线插件开发。此方式实现了无码化的开发，开发复杂度低；适用于二进制消息和 json 格式消息不需经过计算，直接相互转换的场景；但并不适用于二进制消息和 json 格式消息需要计算后再转换的场景。

方式二：离线开发编解码插件。使用 Java 语言开发，开发复杂度相对方式一更高，适用于全部场景。

## 4.2 开发者 Portal 在线插件开发

### 4.2.1 登录开发者 Portal

请使用章节 2.1 获取到的账号和密码登录开发者 Portal。如果初始密码已经修改，请用修改后的密码登录。

开发者 Portal 地址：<https://180.101.147.208:8093>

### 4.2.2 在线完成插件开发

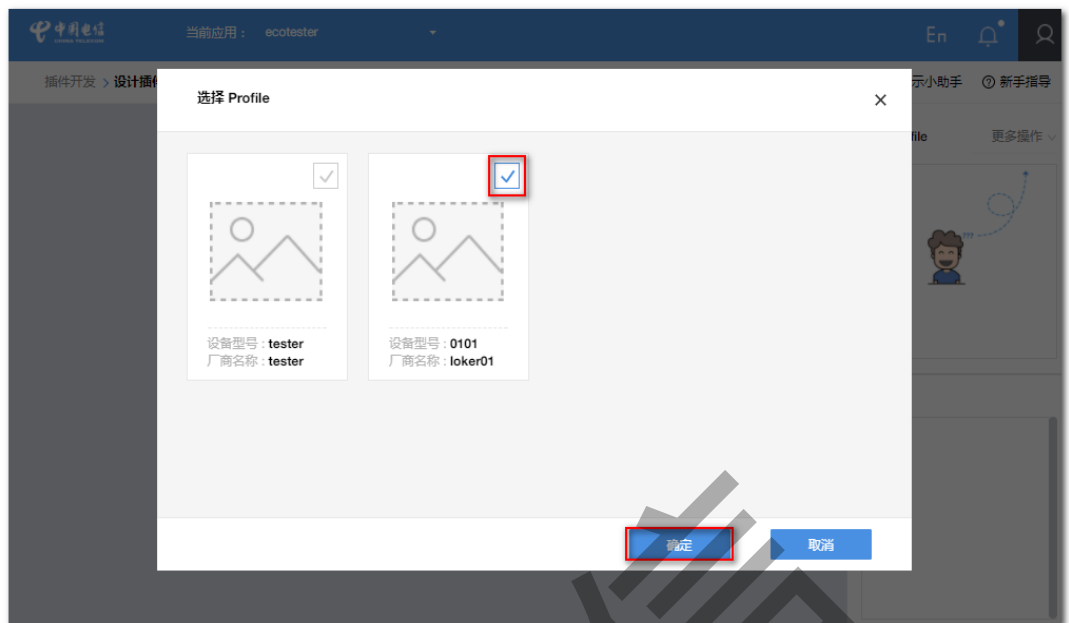
请参考开发者 Portal 使用指导视频完成 Profile 开发。

开发者 Portal 使用指导视频下载地址：

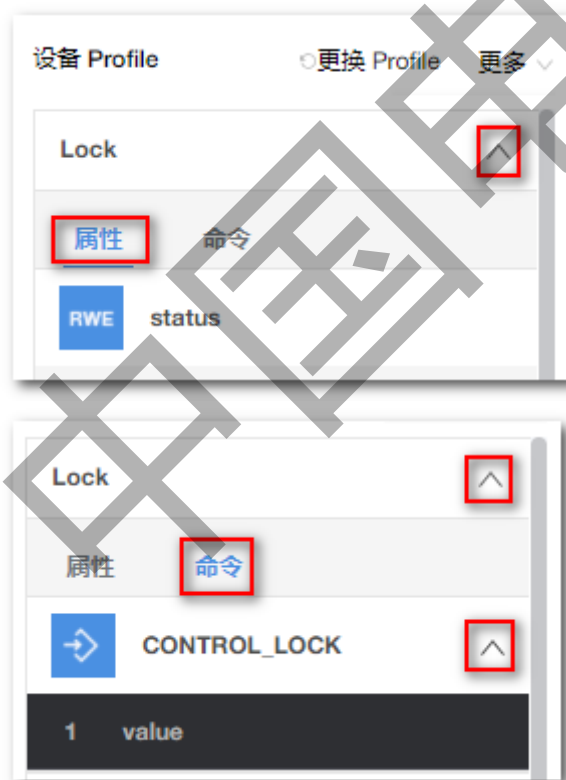
<http://www.tianyiiot.com/downloadZone/downloadZones.html>

大致开发步骤如下：

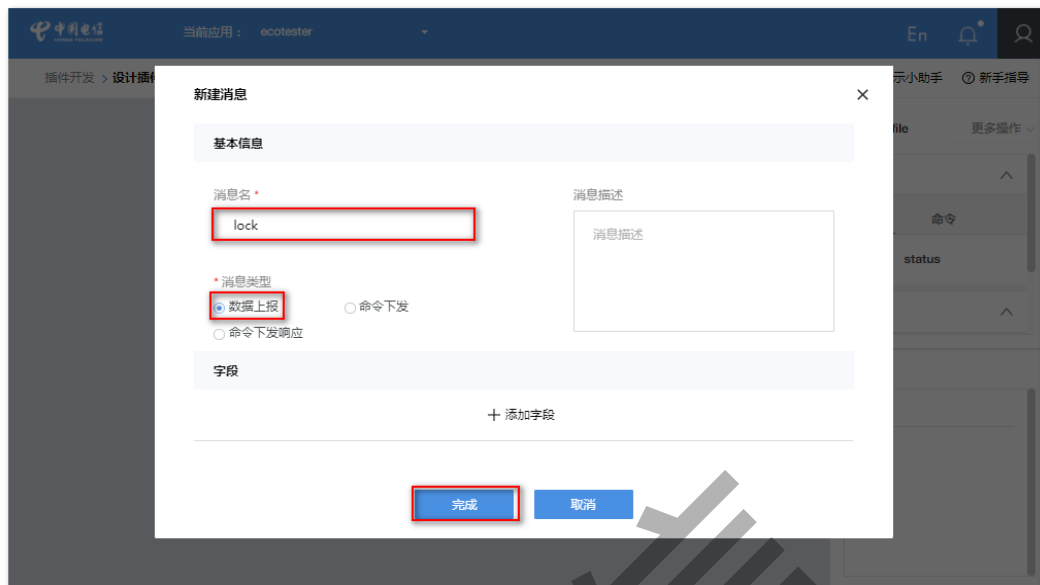
登录开发者 portal->插件开发->插件开发（子菜单）->开始设计->新建插件->选择 Profile（请等待 profile 加载完成后再选择对应 profile）->点击确定



可以在浏览器在右侧看到 profile 中定义的属性和命令



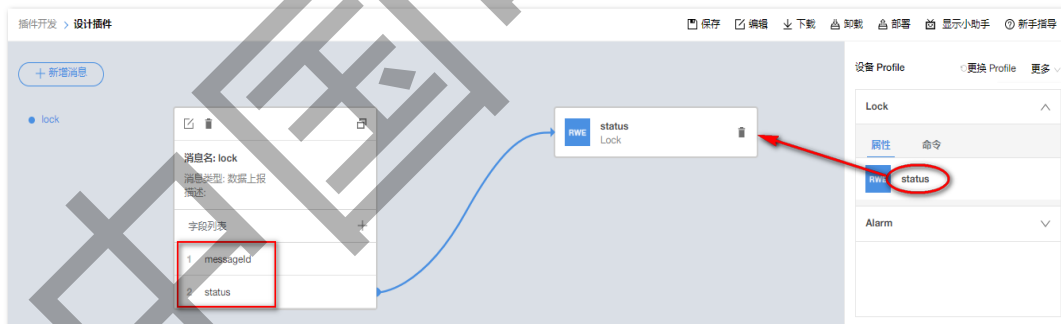
点击新增消息按钮，定义一条数据上报消息（消息名称可自定义）：



根据业务设计，一条码流可以被分成几段，每一段码流代表不同的业务数据，将需要上报的码流段提取出来，做为 **profile** 属性值上报给平台，再推送给应用。一条码流可以上报多个服务下的多个属性，也可以只上报一个服务的一个或者多个属性。

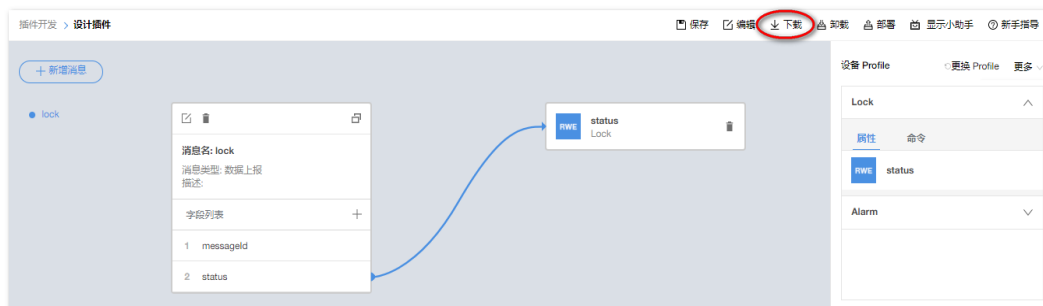
命令下发则与数据上报相反，是把命令字段的值取出来放入码流中。而一个命令下的参数值只能放到一条下发命令的消息中。

具体操作和详细开发过程请直接参考视频。



#### 说明

开发者 Portal 支持下载在线完成的插件，下载后的插件并未经过签名；当需要将开发者 Portal 下载的插件部署到生产环境时，请将插件包签名后和公钥（参考章节 4.3.3 和 4.3.4）一起通过中国电信物联网开放平台支持人员部署到生产环境。



## 4.3 离线开发编解码插件

### 4.3.1 开发编解码插件

编解码插件开发可参考《[中国电信物联网开放平台编解码库开发与升级指南](#)》文档，文档详细介绍了需要实现的 4 个接口，同时附带了样例代码。

### 4.3.2 编解码插件质检

编解码插件的质检是检测编解码是否能够正常使用的关键步骤，请参照编解码插件检查工具说明，来进行操作。

工具下载地址：<http://www.tianyiiot.com/attachment/174/中国电信物联网开放平台编解码插件检测工具.zip>

### 4.3.3 对插件包进行离线签名

当编解码插件开发完后，在安装到平台之前，需要先对插件包进行签名。此时需要下载离线签名工具并进行签名操作。操作步骤如下：

步骤 1 使用浏览器登录 SP Portal。

步骤 2 下载离线签名工具。

1. 单击左侧图标，打开管理页面。
2. 单击左侧导航栏“工具”，在右侧区域单击“下载”，下载离线签名工具。

步骤 3 在下载路径找到压缩包“signtool.zip”，右键菜单中选择“Extract to signtool\”解压缩至文件夹“signtool”。

步骤 4 进入 signtool 文件夹，运行“signtool.exe”。操作界面如下所示。

3. 根据实际情况选择签名算法。

目前提供两种签名算法:

- ECDSA\_256K1+SHA256

## 设置“私钥加密口令”

口令复杂度说明:

- 口令长度至少为 6 个字符

- 至少一个小写字母

- 至少一个大写字母

- 至少一个数字

- 至少一个特殊字符: `~!@\$%^&\*()-\_+=\|[{ }];:","<.>/?` 和空格

可在保存的目录下查看生成的公私密钥文件。

- 公钥文件: public.pem

- 私钥文件：private.pem

步骤 6 对软件包进行数字签名。



说明

离线签名工具只能对.zip 格式的压缩包进行数字签名。

6. 在“软件包数字签名”区域，单击“导入私钥文件”，选择步骤 5.3 中生成的私钥文件，单击“打开”。
7. 在弹出的对话框中，输入步骤 5.2 中设置的口令，单击“确定”。
8. 在“需要数字签名的软件包”区域，选择需要进行数字签名的软件包。单击“打开”。
9. 单击“进行数字签名”。

签名成功后，在原软件包所在目录生成名为“XXX\_signed.XXX”的带签名的软件包。

步骤 7 软件包签名验证。

10. 在“软件包签名验证”区域，单击“导入公钥文件”，选择步骤 5.3 中生成的公钥文件，单击“打开”。
11. 在“需要验签的软件包”区域，选择步骤 6 中生成的名为“XXX\_signed.XXX”的带签名的软件包。单击“打开”。
12. 单击“进行软件包验签”。
  - 验证成功则弹出“验证签名成功!”提示框。
  - 验证失败则弹出“验签异常!”提示框。

----结束

#### 4.3.4 上传公钥及签名后的插件包

编解码插件化能够动态的新增编解码库，这样新增设备接入时，只需要把对应的编解码库动态导入到平台，就能够兼容新增的设备。请将公钥以及签名后的插件包发送给中国电信物联网开放平台支持人员，由中国电信完成公钥和签名后的插件包的上传配置。

# 5 应用接入平台

## 5.1 登陆应用

为了防止应用非法接入中国电信物联网开放平台，应用访问平台时首先需要进行登陆验证，登陆成功后获取访问令牌（accessToken）。登录验证时使用[章节 2.2 创建 APP 应用](#)时获取的 appId 和 secret 进行鉴权。

登录应用的 API 描述请参考《[中国电信物联网开放平台 API 参考 1.3.1](#)》章节 1.2.1.1；

Java 语言 SDK 使用的接口请参考《[中国电信物联网开放平台北向 API Java SDK 说明文档](#)》章节 4.1.1，样例代码请参考章节 8.2 的 AuthenticationTest.java。

 说明

如果多次获取令牌，则之前的令牌失效，最后一次获取的令牌才有效。请勿并发获取令牌。

## 5.2 订阅

为了能接受设备上报的数据，应用服务器需要向中国电信物联网开放平台订阅通知消息。订阅成功后，设备上报数据时，中国电信物联网开放平台会将数据推送到订阅时设置的 callbackurl 上。

订阅的 API 描述请参考《[中国电信物联网开放平台 API 参考 1.3.1](#)》章节 1.2.3.3；

Java 语言 SDK 使用的接口请参考《[中国电信物联网开放平台北向 API Java SDK 说明文档](#)》章节 4.3.5，样例代码请参考章节 8.2 的 DataCollectionTest.java。

## 5.3 注册设备


### 5.3.1 有应用服务器设备注册

所有设备需要先在北向进行注册，才允许连接到平台。通过注册设备，平台会为每个设备分配一个唯一的标识 deviceId，后续应用操作这个设备时都通过 deviceId 来指定设备。另外，还返回 psk 参数（如果用户未指定 psk 参数，平台会随机分配一个参数），南向设备绑定时，如果设备与平台之间走 DTLS 加密通道，则须用到该参数，请保存。在 SP portal 上也可以通过 deviceId 来找到设备。

注册设备的 API 描述请参考《[中国电信物联网开放平台 API 参考 1.3.1](#)》章节 1.2.2.1；



Java 语言 SDK 使用的接口请参考《[中国电信物联网开放平台北向 API Java SDK 说明文档](#)》章节 4.2.1，样例代码请参考章节 8.2 的 DeviceManagementTest.java。


 说明

verifyCode 和 nodeId 需要填写为设备唯一标识；高通芯片设备的唯一标识为 urn:imei:xxxx，xxxx 为 IMEI 号；海思芯片设备的唯一标识为 IMEI 号，其他种类芯片的设备唯一标识请联系模组厂商确认；IMEI 的值请参考 6.1 设备接入一节。timeout 建议填写为 0。

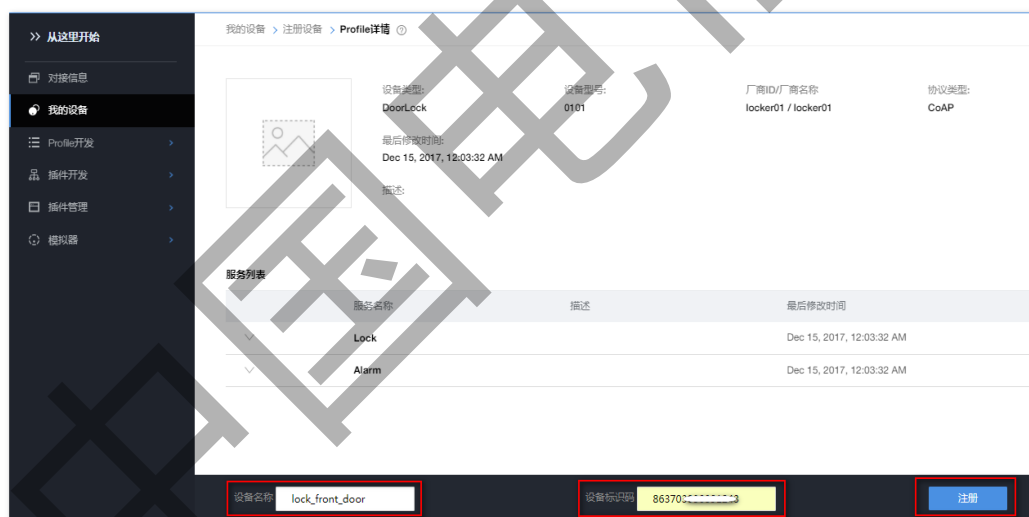
## 5.3.2 无应用服务器设备注册

无应用服务器情况下，厂商可以用开发者 portal 的应用模拟器来进行设备的注册。方便厂商进行联调开发。

登录开发者 portal->我的设备->注册设备->选择需要注册设备的 Profile（即选择一款设备模型）->填写设备名称和设备标识，点击注册按钮即可

 说明

使用开发者 portal 注册设备时，后台已完成了注册设备和设置设备信息两个步骤。




## 5.4 设置设备信息

这一步是为了把设备的厂商、型号、设备类型等信息设置到平台，平台在处理过程中需要这些信息。

设置设备信息的 API 描述请参考《[中国电信物联网开放平台 API 参考 1.3.1](#)》章节 1.2.2.6；

Java 语言 SDK 使用的接口请参考《[中国电信物联网开放平台北向 API Java SDK 说明文档](#)》章节 4.2.4，样例代码请参考章节 8.2 的 DeviceManagementTest.java。

 说明

NB-IoT 业务场景下 manufacturerId、manufacturerName、deviceType、model、protocolType 5 个字段都必须进行设置，并且设置的值要与 profile 中定义的值保持一致。

中国电通信

# 6 设备接入平台

## 6.1 设备接入

设备上电后，使用串口工具配置设备，按照如下AT命令顺序执行即可：

AT 命令	说明
AT+CMEE=1	报错查询
AT+CFUN=0	关机，设置 IMEI 和平台 IP 端口前要先关机
AT+CGSN=1	查询 IMEI，IMEI 即为设备标识，应用注册设备时 nodeId/verifyCode 都需要设置成 IMEI
AT+NTSETID=1, xxxx	xxxx 为 IMEI。如果查询不到可自行设置 IMEI，IMEI 必须是唯一的，不能与其他设备重复，且只能设置一次。IMEI 即为设备标识，应用调用 API 注册设备时，如果设备使用的是海思芯片 nodeId/verifyCode 都需要设置成 IMEI，如果设备使用的是高通芯片 nodeId/verifyCode 都需要设置成 urn:imei:IMEI。
AT+NCDP=180.101.147.115, 5684	设置对接的平台 IP 端口，5683 为非加密端口，5684 为加密端口
AT+CFUN=1	开机
AT+NBAND=5	设置频段
AT+CGDCONT=1, "IP", "CTNB"	设置核心网 APN，APN 与设备的休眠、保活等模式有关，需要与运营商确认。电信物联网开放平台 APN 见下方的图表。
AT+CGATT=1	入网
AT+CGPADDR	获取终端 IP
AT+NMGS=2, 0001	发送上行数据，第 1 个参数为字节数，第 2 个参数为上报的 16 进制业务码流
AT+NQMGR	接收下行数据
AT+NMGR	读取数据

使用非加密端口接入时，只需要使用原生的 CoAP 协议进行传输，同时将对接平台设置为 5683 即可。

电信物联网开放平台 APN 如下图所示：

序号	APN名称	PSM Active Timer	eDRX	
		T3324	T <sub>eDRX</sub>	T <sub>PTW</sub>
1	psmA.eDRX0.ctnb	2s	不启用	
2	psmC.eDRX0.ctnb	60 s	不启用	
3	psmF.eDRXC.ctnb	180 s	20.48 s	10.24 s
4	psm0.eDRXH.ctnb	不启用PSM	655.36 s	10.24 s
5	psm0.eDRXD.ctnb		40.96 s	
6	psm0.eDRXC.ctnb		20.48 s	
7	psm0.eDRX0.ctnb		不启用	
8	ue.prefer.ctnb	终端控制		
9	Ctnb	2s	不启用	

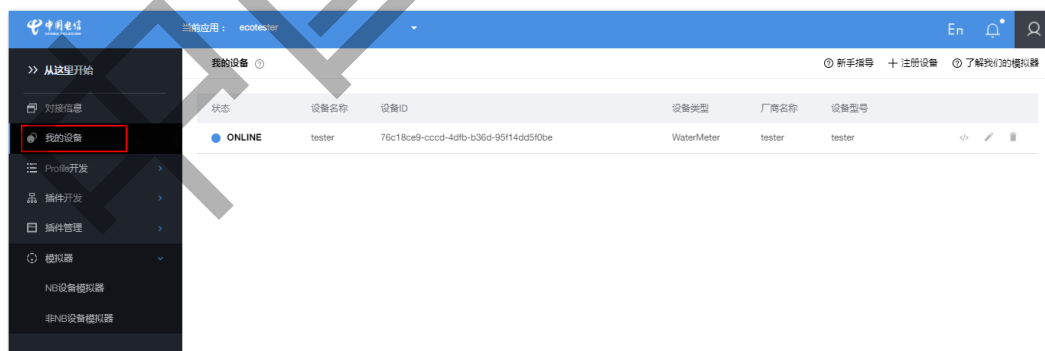
## 6.2 设备上线

### 6.2.1 真实设备

完成这一步时，设备已经可以接入到平台。配置好网络，开启设备，观察设备是否成功接入到平台。

登陆开发者 Portal，在我的设备列表页面，查看设备状态，字段设备 ID 即为在第三步里注册设备时生成的设备 ID，状态字段表示设备的在线状态，如果状态是在线（online）表示设备已经成功的接入到平台，接着就可以接收设备的数据。

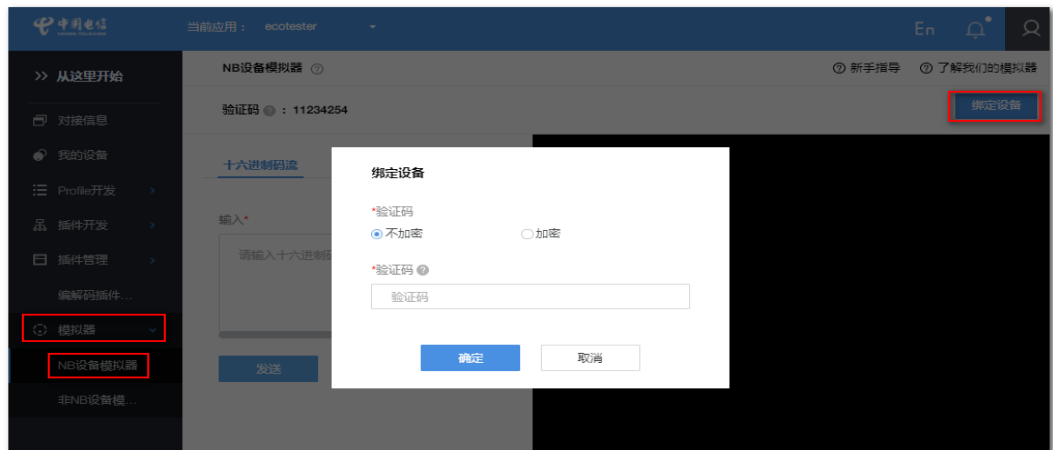
图6-1 查看设备列表



### 6.2.2 模拟设备

联调过程中也可用设备模拟器来辅助验证。

登录开发者 portal，选择模拟器->NB 设备模拟器->绑定设备，在弹出的对话框中输入验证码（即设备标识、IMEI），点击确定即可。



绑定成功后就可以直接使用设备模拟器发送业务码流。



设备数据上报并解析成功后可以在开发者 portal->我的设备->点击要查看的设备进入设备详情->历史数据中查看设备数据：

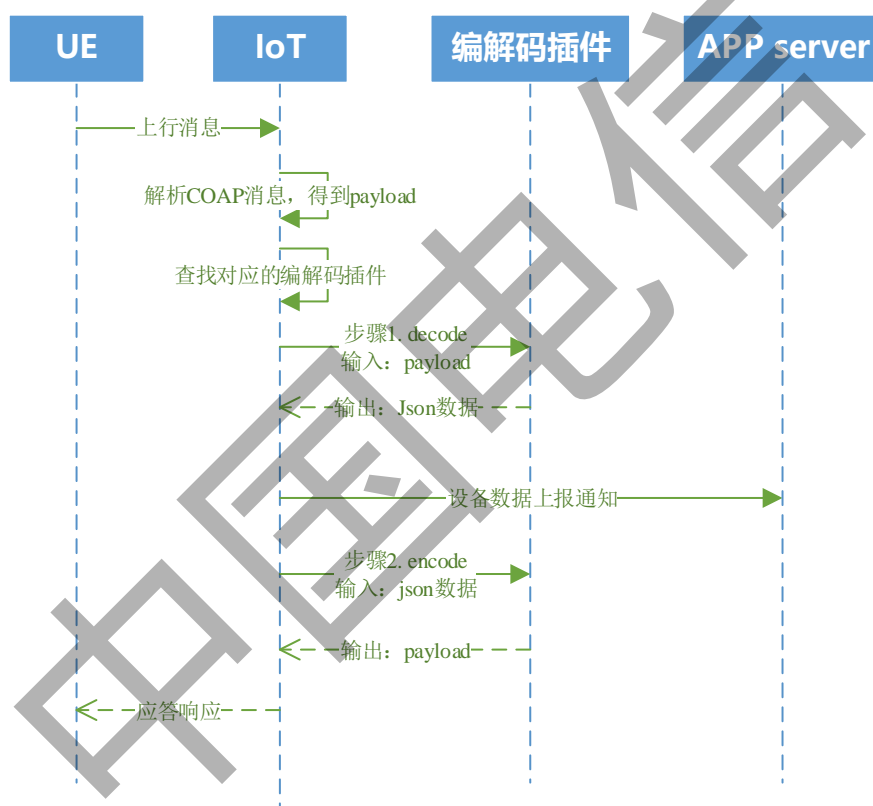


# 7

## 业务数据上报和业务消息下发流程

### 7.1 接收设备上报的数据

图7-1 上行消息处理流程



前面已经讲到应用向平台订阅数据上报通知消息。设备上报数据时，平台会把数据推送给应用订阅的地址上。

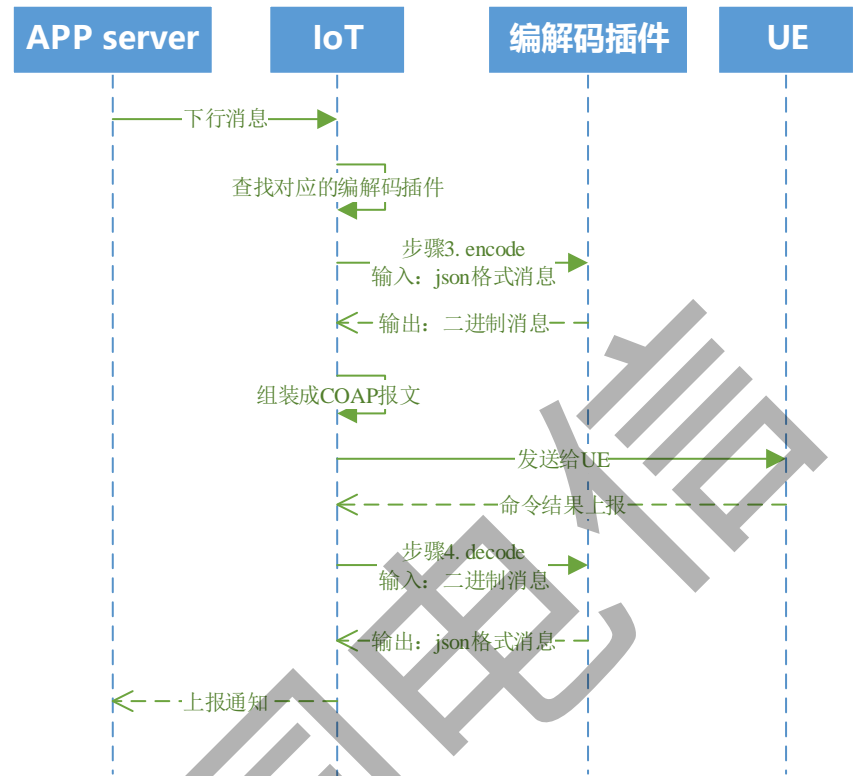
```
POST
https://10.3.3.5:9999/app/notify
Body:
{
  "notifyType": "deviceDatasChanged",
  "requestId": null,
  "deviceId": "b8b92cc7-2622-4f27-a24b-041ab26f0b80",
  "gatewayId": "b8b92cc7-2622-4f27-a24b-041ab26f0b80",
  "services": [
    {
```

```
        "serviceId": "Brightness",
        "serviceType": "Brightness",
        "data": { "brightness": 50},
        "eventTime": "20170214T170220Z"
    },
    {
        "serviceId": "Electricity",
        "serviceType": "Electricity",
        "data": {
            "voltage": 218.90001,
            "current": 800,
            "frequency": 50.1,
            "powerfactor": 0.98
        },
        "eventTime": "20170214T170220Z"
    },
    {
        "serviceId": "Temperature",
        "serviceType": "Temperature",
        "data": { "temperature": 25},
        "eventTime": "20170214T170220Z"
    }
}
```

上面消息里的 services 的 Brightness, Electricity 及 Temperature 服务数据就是经过编解码插件解析出来的，其为标准 json 格式的数据，字段和产品设备模型定义的一致。

## 7.2 向设备发送消息

图7-2 下行消息处理流程



应用向设备发送消息使用“1.2.4 信令传送”接口，请参考《[中国电信物联网开放平台 API 参考 1.3.1](#)》。

Java 语言 SDK 使用的接口请参考《[中国电信物联网开放平台北向 API Java SDK 说明文档](#)》章节 4.4.1，样例代码请参考章节 8.2 的 `SignalDeliveryTest.java`。

向设备下发上文模型中对 `Temperature` 服务设置定义的命令 `SET_TEMPERATURE`

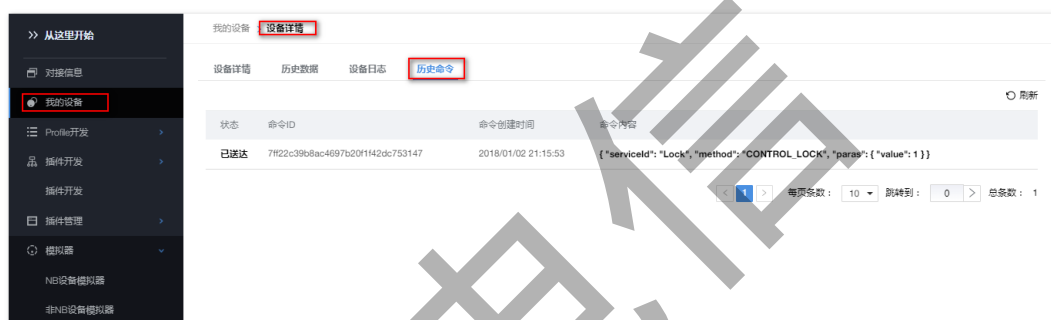
```
POST
https://server:port/iocm/app/cmd/v1.3.0/devices/{deviceId}/commands
app_key: *****
Authorization:Bearer *****
Content-Type:application/json
Body:
{
  "command": {
    "serviceId": "Temperature",           // Temperature 服务名
    "method": " SET_TEMPERATURE ",       // Temperature 服务命令名
    "paras": {
      "value":30                          //命令参数
    }
  }
}
```



```
response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "requestId": "de651c90331c4d11ba94b8cef3810efe",
  "commandId": "76553ad00df34a88974fbcefd42d510",
  "commandStatus": "SENT"
}
```

可以在开发者 portal->我的设备->点击对应的设备->历史命令页面，查看命令下发的情况。

图7-3 命令下发状态查看



当设备收到命令，模组会自动回复一个 ACK（2.04 Changed）命令应答（此为 CoAP 协议层的应答），命令的状态由“已发送”状态变为“已送达”状态。

当设备执行完命令，上报一条命令执行成功结果的码流并经插件 decode 解析，上报 deviceRsp 时，命令状态由“已送达”变为“成功”。

#### 说明

命令的执行结果处理只能使用线下开发编解码插件的方式才能处理。开发者 portal 在线开发目前仅支持数据上报和命令下发的处理。

如果设备需要上报命令的执行结果，建议使用平台给的 mid 来标识命令和命令执行结果。在命令下发时，encode 函数的入参中平台会给每个命令配置一下 mid（值为 int 类型，从 1 开始递增），插件将 mid 的值加到命令码流里（转成 4 位 16 进制码流，不够 4 位需要在前面补 0），下发给设备。设备执行完命令后，再将 mid 与命令结果一起报上来。在 decode 函数中将上报的 mid 转为 int 类型并返回。具体可参考 4.3 离线开发编解码插件中提供的文档和插件样例。

图7-4 命令执行成功



#### 说明

在命令经过平台发送后，在一定时间内，如果设备没有返回 ACK（2.04 Changed）命令应答，则命令状态会变成“超时”。

中国电信用

# 8 命令下发

平台提供两种命令下发机制：

**立即下发：**平台立即发送收到的命令，如果设备不在线或者设备没收到指令则下发失败。立即下发适合对命令实时性有要求的场景，比如路灯开关灯，燃气表开关阀。使用立即下发时，应用需要自己保证下发的时机。

**缓存下发：**平台收到命令后放入队列。在设备上线的时候，平台依次下发命令队列中的命令。缓存下发适合对命令实时性要求不高的场景，比如配置水表的参数。缓存下发平台根据设备的省电模式进行不同处理。

应用服务器向中国电信物联网开放平台下发命令时，携带参数 `expireTime`（简称 TTL，表示最大缓存时间）。如果不带 `expireTime`，则默认 `expireTime` 为 48 小时。

`expireTime=0`：命令立即下发。

`expireTime>0`：命令缓存下发。

命令包含的状态：取消，已发送，已送达，过期，超时。

 说明

只有缓存命令在发送前才能使用北向接口设置为取消状态。立即下发命令不能取消。

## 8.1 命令立即下发

处理流程：

**步骤 1** 应用服务器调用北向接口立即下发命令，参数 `expireTime` 传 0 表示立即下发，样例：

```
Method: POST
request: https://server:port/iocm/app/cmd/v1.3.0/devices/{deviceId}/commands
app key: {appId}
Authorization:Bearer *****
Content-Type:application/json
Body:
{
```

```

    "requestId": "*****",
    "command": {
      "serviceId": "*****",
      "method": "*****",
      "paras": {
        "paraName1": "paraValue1",
        "paraName2": "paraValue2"
      }
    },
    "callbackUrl": "http://127.0.0.1/cmd/callbackUrl",
    "expireTime": 0
  }

```

步骤 2 平台收到后，调用插件编码。

步骤 3 平台将命令下发给对应插件的 encode 函数，encode 入参样例如下：

```

{
  "identifier":0,
  "msgType":"cloudReq",
  "serviceId":"NBWaterMeterCommon",
  "mid":2016,
  "cmd":"SET_TEMPERATURE_READ_PERIOD",
  "paras":{"value":4},
  "hasMore":0
}

```

步骤 4 NB 模组收到命令时，回 CoAP 协议的 ACK 消息（注：ACK 消息对设备应用不可见）

步骤 5 平台收到 ACK 后，认为命令送达设备。向 NA 推送送达通知。消息样例：

```

{
  "deviceId" : "6213fa11-68d6-4457-aa98-871a25c152c1",
  "commandId" : "30d188e1-2816-41a4-989f-1797c74b1745",
  "result" : {
    "resultCode" : "DELIVERED",
    "resultDetail" : null
  }
}

```

步骤 6 UE 执行完命令，如果需要上报命令执行结果，则上报执行结果

步骤 7 平台调用插件解码，解码输出样例： 其中 mid 参数表示命令标识，应该和 encode 输入的一致

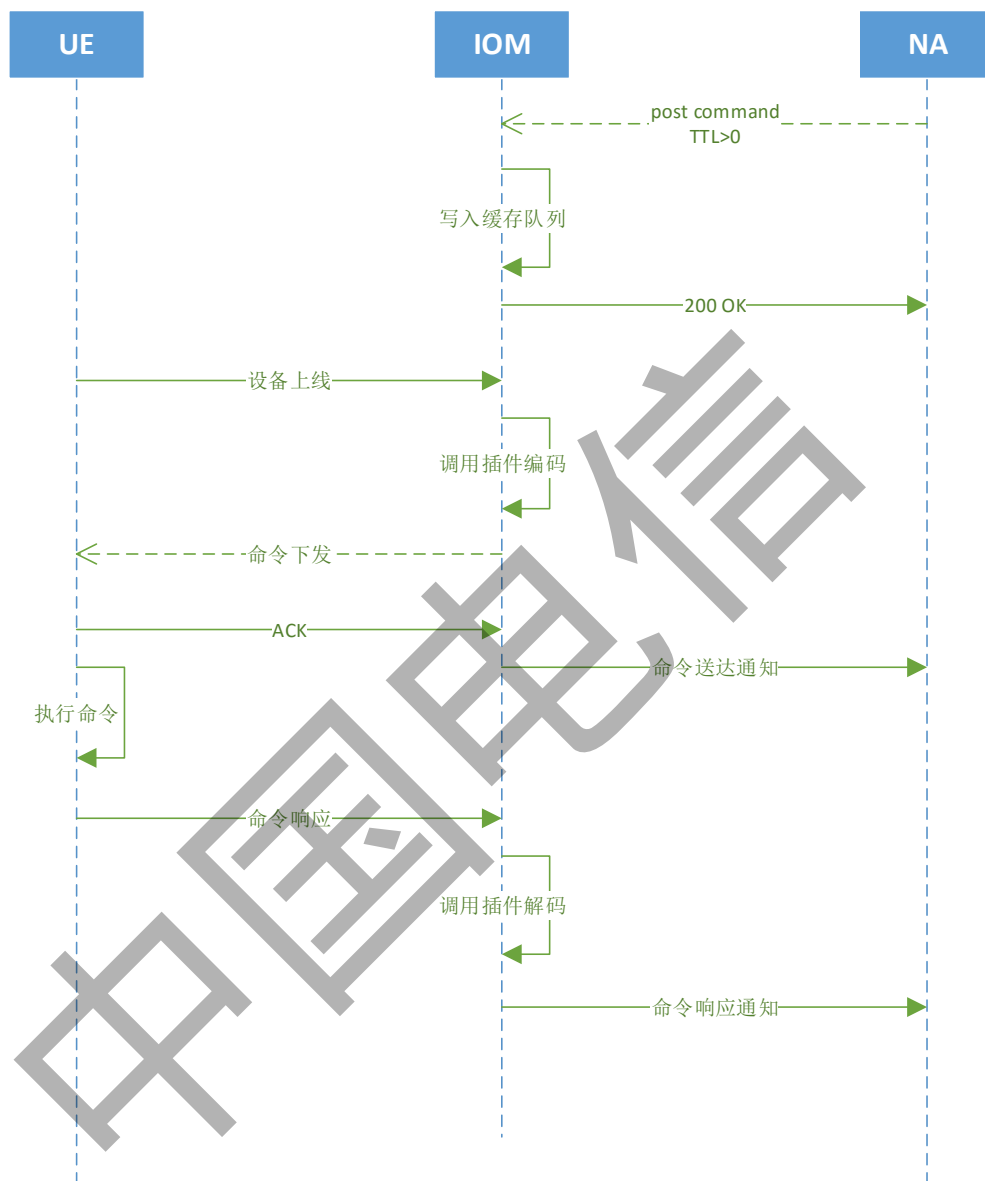
步骤 8 平台根据 mid 查找命令，并给 NA 上报命令响应通知，如果没有 mid，那么平台不去匹配命令来更新命令成功或失败的状态，推送给 NA 的 commandID 为 null。插件收到命令执行结果后，decode 函数解析完成上报的命令应答 json 样例：

```

{
  "identifier":"123",
  "msgType":"deviceRsp",
  "mid":2016
  "errcode":0,
  "body" :{ "result":0}
}

```

## 8.2 命令缓存下发



- 步骤 1** NA 下发命令时，`expireTime` 传大于 0 的值表示缓存下发，平台把命令写入缓存队列，返回 200 OK。
- 步骤 2** 平台会根据设备省电模式决定何时下发缓存命令：对 PSM 模式设备，平台等待设备上报数据时下发缓存命令；对其他模式设备，则主动发送缓存命令到设备。
- 步骤 3** 平台发送缓存命令时，按照写入队列的先后顺序串行下发，前一个命令送达设备时（即收到 NB 模组发送的 ACK 报文）开始下发后一个缓存命令。
- 步骤 4** 平台对缓存命令的最大缓存时间有限制，由参数 `expireTime` 指定。当命令在平台缓存的时间超过 `expireTime` 时，命令将从缓存队列移除，不再进行发送处理。
- 步骤 5** 缓存命令的其他处理过程和立即命令相同。

## 9 更多功能

---

上述介绍了基本功能的使用，如果需要使用更多功能，请参考《[中国电信物联网开放平台 API 参考 1.3.1](#)》和《[中国电信物联网开放平台北向 API Java SDK 说明文档](#)》。

## 10 测试环境测试

---

### 10.1 测试申请

企业客户首先需要本地测试；本地测试完成后，可以通过“天翼物联产业联盟”公众号申请开放实验室测试。

### 10.2 开放实验室测试

在中国电信物联网开放实验室，中国电信测试人员配合企业客户一起完成测试，并且输出测试报告。

# 11 生产环境接入

---

## 11.1 账号申请

当测试平台验证通过后，企业客户可以通过中国电信客户经理提出申请接入中国电信物联网开放平台生产环境。账号申请通过后，企业客户将通过邮件收到账号和初始密码。

## 11.2 生产环境接入

生产环境接入流程与测试环境接入流程和方法基本一致，在此不再详述。

**需要注意的是：**

- 测试环境定义的 profile 和开发的插件都可以导出，并上传到生产环境
- 如果使用开发者 portal 在线开发，导出的插件是未签名的插件包，需要签名后和公钥一起配合使用才能上传到生产环境
- 模拟器不能替代真实的应用和设备，在生产环境，需要通过 API 和真实终端设备完成设备接入平台

# 12 参考文档

---

## 12.1 中国电信物联网开放平台编解码库开发与升级指南



中国电信物联网开  
放平台V100R001C

## 12.2 中国电信物联网开放平台 API 参考 1.3.1



中国电信物联网开  
放平台API参考1.3.1

## 12.3 中国电信物联网开放平台北向 API Java SDK 说明文档



中国电信物联网开  
放平台\_北向API Ja